



## UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/564,210	01/09/2006	Hans-Joachim Beck	2003P09791WOUS	7538
22116	7590	02/05/2009	EXAMINER	
SIEMENS CORPORATION INTELLECTUAL PROPERTY DEPARTMENT 170 WOOD AVENUE SOUTH ISELIN, NJ 08830			PROCTOR, JASON SCOTT	
ART UNIT	PAPER NUMBER			
		2123		
MAIL DATE	DELIVERY MODE			
02/05/2009	PAPER			

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	<b>Application No.</b> 10/564,210	<b>Applicant(s)</b> BECK, HANS-JOACHIM
	<b>Examiner</b> JASON PROCTOR	<b>Art Unit</b> 2123

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED. (35 U.S.C. § 133).

Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

1) Responsive to communication(s) filed on 26 January 2009.

2a) This action is FINAL.      2b) This action is non-final.

3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

4) Claim(s) 11-13 and 15-23 is/are pending in the application.

4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.

5) Claim(s) \_\_\_\_\_ is/are allowed.

6) Claim(s) 11-13 and 15-23 is/are rejected.

7) Claim(s) \_\_\_\_\_ is/are objected to.

8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

9) The specification is objected to by the Examiner.

10) The drawing(s) filed on \_\_\_\_\_ is/are: a) accepted or b) objected to by the Examiner.  
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a) All    b) Some \* c) None of:

1. Certified copies of the priority documents have been received.
2. Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

1) Notice of References Cited (PTO-892)

2) Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) Information Disclosure Statement(s) (PTO/1449B)  
 Paper No(s)/Mail Date \_\_\_\_\_

4) Interview Summary (PTO-413)  
 Paper No(s)/Mail Date \_\_\_\_\_

5) Notice of Informal Patent Application

6) Other: \_\_\_\_\_

## **DETAILED ACTION**

Claims 10-13 and 15-22 were rejected in the Office Action entered on 7 November 2008.

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 6 January 2009 has been entered.

The 6 January 2009 submission has amended claims 11, 12, 13, 15-19, and 22; canceled claim 10; and presented new claim 23. Claims 11-13, and 15-23 are pending in this application.

Claims 11-13 and 15-23 are rejected.

### ***Response to Arguments – 35 USC § 103***

1. In response to the previous rejection of claims 10-13 and 15-22 under 35 U.S.C. § 103(a) as being unpatentable over Heile in view of Hoskins, Applicants argue primarily that:

In the office communication on page 5, par.2, Examiner cites Heile col. 18, lines 35-65 and FIG 13 as teaching saving of references on the programming device, wherein the references indicate which project design blocks are to be copied from the library to the programming device. However the cited lines of Heile do not teach a reference on a programming device indicating which design blocks are to be copied. The cited lines only describe a request by a user for a given design block.

The Examiner respectfully traverses this argument as follows.

In column 18, lines 35-65 and FIG. 13, Heile teaches a method of copying the design blocks to be copied from the library to the programming device. Heile also teaches saving a reference on each programming device indicating which project design blocks are to be copied from the library to the programming device [*"Project database 100 may be implemented and*

*stored on a wide variety of computers in a wide variety of manners. By way of example, project database 100 may be stored on a single computer system, or preferably project database 100 is implemented on a number of computer systems, each being used by a single engineer. Project database 100 includes a global work space 102 that is connected electronically via a network connection, cable, computer hardware or other similar connections 103 to a User One work space 104, a User Two work space 106, and a User Three work space 108.*" (column 8, lines 35-50)]. Further, Heile expressly teaches storing a reference on each programming device indicating which project design blocks are to be copied from the library to the programming device [*User One work space 454 includes downloaded versions of project source files 480, processing results 482, a local assignment database 484 and a local assignment file 486. Local database 484 is a binary database that may be implemented in the same manner as global database 460. Database 484 contains in record form the assignments that User One is currently viewing or editing.*" (column 17, line 65 – column 18, line 20)].

2. Applicants further argue that:

On page 5, par. 5, Examiner equates Heile's global central workspace with Applicants' programming devices 13-15. However, Applicants' programming devices 13-15 are local devices (par. 18 line 7), so this aspect of the rejection is unclear.

The Examiner respectfully traverses this argument as follows.

Heile teaches transferring the copied design blocks to the programming device, by the engineering system; and storing the transferred design blocks on the programming device (FIG. 13). The Examiner does not equate Heile's "global central workspace" with Applicants' programming devices 13-15.

3. Applicants further argue that:

Heile's cited FIG. 13 does not teach that a given design block is included in multiple part-projects and that a given design block is transferred to multiple programming devices for concurrent use (Applicants' FIG 1). In fact, Heile's method of FIG 13 specifically prevents this, by teaching that a user cannot check-out a file if it is locked for another user.

The Examiner respectfully traverses this argument as follows.

Heile teaches that a second user cannot check-out a file **if it is locked** for another user. However, Heile teaches that a given design block is part of more than one part project, and a copy of the given design block is transferred to more than one of the programming devices for concurrent use [*"Any number of local work spaces contain downloaded versions of any of the project source files."* (abstract); FIG. 10 showing "A Version 2.0" in workspace "User One 354," "User Two 356," and "User Five 362"].

4. Applicants further argue that:

Heile does not teach that two part-projects on respective local programming devices are functionally linked for operational data transfer therebetween as now claimed per Applicants' par. 16, lines 25-30.

The Examiner respectfully traverses this argument as follows.

Heile teaches that two part projects on respective local programming devices are functionally linked for operational data transfer therebetween [*"In this example, User One has*

*transferred record X 488 to the local work space and has placed record X 488 into a default state so that any changes to record X in the global database 460 will automatically be retrieved to update the user's local database 484.*" (column 17, line 65 – column 18, line 20)]. That is, Heile teaches that a part project on one programming device receives an operational data transfer to update certain files after those certain files have been updated on a second programming device.

5. Applicants further argue that:

Examiner cites Heile's step 516 (FIG 13) as corresponding to Applicants' required responses from all programming devices before a project design block on any programming device is updated from the central library. However step 516 controls updating a file in the central library from a programming device, not updating a file on a programming device from the central library. Furthermore, it does not require a response from any other user. Steps 516-520 either allow or block updating of a file in the central library depending on whether the user submitting the check-in update has locked the file or not.

The Examiner respectfully traverses this argument as follows.

Heile teaches replacing at least one project design block on each local programming device by the corresponding block stored in the library if and only if the user request is accepted by the users of all programming devices [*"File A' 266 in user work space 254 is in a default state meaning that the file is read-only by the user and that any new versions of file A 256 in the global work space are automatically retrieved and transferred to the user work space."* (column 14, lines 23-39); *"File B' 268 is in a locked state meaning that the user has transferred the latest version of file B 258 from the global work space in order to edit it. ... A locked state supports users who wish to exclusively edit a file that eventually will be shared with other users."* (column 14, lines 40-51)]. That is, according to a **user request** to check out a file in **default** or **locked** state determines whether a project design block on each local programming device is replaced by

the corresponding block stored in the library. A user may accept that request (default state) or refuse the request (locked state).

6. Applicants further argue that:

Regarding claim 22, Examiner cites Heile's method of FIG 14, but this method never prompts all users for acceptance of an update. Instead, whether to update a user's local file or not is determined automatically in Heile. Thus, a user may be updated unexpectedly without consent, or a user may not be updated, causing loss of version synchronization among the users.

The Examiner respectfully traverses this argument as follows.

The question of updating a user's local file in Heile is not determined "automatically" as alleged by Applicants' but instead is determined at the time of check-out according to the file states shown in FIG. 11. A user's local file is never "updated unexpectedly without consent" because there is no file state in FIG. 11 corresponding to that event. If a local file is checked out in the "default state," that file may be updated unexpectedly (when another user has made changes) but that update is made with the user's **explicit consent** by using the "default state". If a user does not consent to any such update, that user may check out a file using the "owned-read only" state, which does not consent to any unexpected updates. Further, as taught by Heile, proper use of the file states shown in FIG. 11 will **enhance and preserve** synchronization among the users (column 12, line 60 - column 16, line 60) rather than disrupt synchronization as alleged by Applicants.

Applicants' arguments have been fully considered but have been found unpersuasive. The previous rejections under 35 U.S.C. § 103(a) as unpatentable over Heile in view of Hoskins are maintained or applied to the new claims as appropriate.

#### *Claim Rejections - 35 USC § 103*

The following is a quotation of 35 U.S.C. § 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. § 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

7. Claims 11-13 and 15-23 are rejected under 35 U.S.C. § 103(a) as being unpatentable over US Patent No. 6,298,319 to Heile et al. (hereafter referred to as Heile) in view of US Patent No. 6,108,662 to Hoskins et al. (hereafter referred to as Hoskins).

Regarding claim 23, Heile teaches a method of designing or configuring an electronic device for controlling a plant, the method comprising:

providing an engineering system comprising a central management unit connected to a plurality of local programming devices by a bus system (Heile, FIG. 1);

the central management unit dividing a project for designing or configuring the process control system into a plurality of part projects for execution on a respective plurality of the programming devices, each part project comprising at least one project design block, and at least some of the part projects comprising plural project design blocks [*"To achieve the foregoing, and in accordance with the purpose of the present invention, a technique and system are disclosed that allow multiple engineers to collaborate in a work group on an electronic design, such as on a ... programming logic device, etc."* (column 3, lines 5-20); *"Advantageously, instead*

*of a complex PLD design being broken up into multiple, smaller PLDs that are assigned to one engineer each, multiple engineers can efficiently work on a very complex design that can be implemented on a single, large capacity PLD. Each engineer is allowed to work on a portion of the overall design." (column 3, lines 19-37); "A project contains one or more hierarchies of design entities and each design hierarchy tree has a root entity, which is the topmost design entity in that hierarchy tree (the top-level functional block). Other design entities in the design hierarchy tree are called child entities." (column 5, lines 48-63)];*

storing the project design blocks in a central library of a memory unit of the central management unit, the programming devices configured to store local copies of such project design blocks required for designing or configuring the process control system [(FIG. 3); "FIG. 3 illustrates symbolically one embodiment of a project database 100. Project database 100 is a database of source files and processing results utilized by multiple engineers in the context of creating a design for a PLD. ... Project database 100 includes a global work space 102 that is connected electronically via a network connection, cable, computer hardware or other similar connections 103 to a User One workspace 104, a User Two work space 106, and a User Three work space 108." (column 8, lines 35-50); "With multiple engineers, each design project in the system includes a global work space and a work space for each user. ... When a user views or edits a file, he does so in his own work space. ... The global work space directory, where the central database is stored, has the currently checked in version of each file, along with the currently checked in version of the global processing operation results, or basis." (column 9, lines 11-34)];

saving a reference on each programming device indicating which project design blocks are to be copied from the library to the programming device [*"Project database 100 may be implemented and stored on a wide variety of computers in a wide variety of manners. By way of example, project database 100 may be stored on a single computer system, or preferably project database 100 is implemented on a number of computer systems, each being used by a single engineer. Project database 100 includes a global work space 102 that is connected electronically via a network connection, cable, computer hardware or other similar connections 103 to a User One work space 104, a User Two work space 106, and a User Three work space 108."* (column 8, lines 35-50); *"User One work space 454 includes downloaded versions of project source files 480, processing results 482, a local assignment database 484 and a local assignment file 486. Local database 484 is a binary database that may be implemented in the same manner as global database 460. Database 484 contains in record form the assignments that User One is currently viewing or editing."* (column 17, line 65 – column 18, line 20)];

copying the design blocks to be copied from the library to the programming device based on the reference, by the engineering system [*"User work space 106 includes a downloaded file D' 120 and processing results 122. In this scenario, the user has downloaded file D from the global work space and has edited the file in order to produce file D'."* (column 10, lines 43-67); etc.];

transferring the copied design blocks to the programming device, by the engineering system [(FIG. 13); (column 10, lines 43-67); etc.]; and

storing the transferred design blocks on the programming device [(FIG. 13); (column 10, lines 43-67); etc.];

wherein a given design block is part of more than one part project, and a copy of the given design block is transferred to more than one of the programming devices for concurrent use [(FIG. 10, “A Version 2.0” shown in several User work spaces)]; and

wherein part-projects on at least two different programming devices are functionally linked for operational data transfer therebetween [*In this example, User One has transferred record X 488 to the local work space and has placed record X 488 into a default state so that any changes to record X in the global database 460 will automatically be retrieved to update the user's local database 484.*” (column 17, line 65 – column 18, line 20)].

Heile does not expressly disclose Applicants’ claimed **intended use** that the “project” is “a process control system of actuators, sensors, programmable controllers, and operating and observation stations,” and that “project design blocks comprise software objects representing operating and observation systems, input and output modules, the actuators, the sensors, and software blocks for creating control programs for the programmable controllers.”

Hoskins teaches a system software solution for controlling an enterprise comprising one or more components for controlling one or more aspects of an industrial environment with code that creates a database of components, each of the components containing control, diagnostic and resource information pertaining to enterprise resources utilized in the industrial environment (abstract).

Hoskins teaches that “programmable controllers are well-known systems for operating industrial equipment, such as assembly lines and machine tools, in accordance with a stored program. In these controllers, a stored program is executed to examine the condition of specific sensing devices on the controlled equipment, and to energize or de-energize selected operating

devices on that equipment contingent upon the status of one or more of the examined sensing devices." (column 1, lines 25-37)

Hoskins teaches that "In the automotive industry, various automotive parts are conveyed along machine lines consisting of many consecutive workstations... The machine line may consist of any number of different stations, each station performing a different procedure on the unfinished block." (column 1, lines 38-54).

Hoskins teaches that "**In this type of system, a programmable controller would receive inputs from all of the various tools at all of the workstation and would provide activating output signals to synchronize the machine operation.**" (column 1, lines 55-64).

Hoskins teaches **actuators and sensors** (column 3, lines 1-14).

To solve problems in the prior art of programming an industrial control system, Hoskins teaches that "Manufacturing customers have long desired an integrated environment for generating an initial design schematic specifying a functional description of a manufacturing environment without the need for specifying product and manufacturing details. The system is provided with a designer studio that utilizes a common database of pre-architected modules to integrate a total system solution for the enterprise." (column 4, lines 20-29).

Therefore, Hoskins teaches a process control system of actuators, sensors, programmable controllers, and operating and observation stations, and Hoskins teaches that when programming the programmable controllers, project design blocks comprise software objects representing operating and observation systems, input and output modules, the actuators, the sensors, and software blocks for creating control programs for the programmable controllers.

Heile and Hoskins are analogous art because both are directed to provided a system for programming a programmable controller.

It would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to combine the teachings of Heile with Hoskins as expressly motivated by Hoskins to provide an easier to use system for programming the controllers (column 4, lines 20-49). Hoskins teaches that the inputs and outputs of the programmable controller correspond to sensors, actuators, and other tools at the workstations (shown above, column 1, lines 55-64).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention to combine the teachings of Heile and Hoskins to arrive at the invention specified in claim 23.

Regarding claim 11, Heile teaches that at least one of the programming devices has at least one of the project design blocks required for designing or configuring the project before the copied design blocks are transferred, the method further comprising:

Comparing a software version of the at least one project design block to a software version of a corresponding project design block stored in the library and replacing upon a request by a user the at least one project design block with a copy of the corresponding project design block stored in the library, if the software version of the at least one project design block is older than the software version of the corresponding project design block stored in the library [*"By way of example, an EDA tool may have a complete version control system custom built into the central database... Also, an EDA tool may have built in support for several of the major and most popular version control systems such as PVCS, RCS and SCCS."* (column 13, lines 4-17);

*"With this linear list, it is possible to bring an old version to the end of the list and thus, undo all the edits between the most recent version and the old version."* (column 13, lines 18-30)].

Regarding claim 12, Heile teaches erasing at least one of the references on said one programming device, and blocking the replacement of such project design block corresponding to the erased reference [*"When version control is being used, files from one user's point of view may have the following states: default, locked, owned-write and owned-read only, although other file states are possible."* (column 14, lines 5-9); *"Having a file in this owned-read only state prevents any accidental writing to the file, and is useful if a user does not wish a file to be automatically updated."* (column 14, line 64, et seq.)].

Regarding claim 13, Heile teaches that the user request is displayed on each programming device [*"Assignment file 462 is an ASCII text file located on disk in one embodiment. This file 462 contains a user readable version of the current state of assignments of the project, and may show the hierarchical path for each assignment."* (column 17, lines 54-64)]; and

The at least one project design block is replaced by the corresponding project design block stored in the library only if the user request is accepted by the users of all programming devices [*"File A' 266 in user work space 254 is in a default state meaning that the file is read-only by the user and that any new versions of file A 256 in the global work space are automatically retrieved and transferred to the user work space."* (column 14, lines 23-39); *"File B' 268 is in a locked state meaning that the user has transferred the latest version of file B 258*

*from the global work space in order to edit it. ... A locked state supports users who wish to exclusively edit a file that eventually will be shared with other users.”* (column 14, lines 40-51)]

Claims 15-17 and 19 recite a system for performing the method of claims 23 and 11-13. A system is disclosed by Heile [FIG. 1]. Claims 15-17 and 19 are rendered obvious over Heile in view of Hoskins for rationale similar to that shown above regarding claims 23 and 11-13.

Claim 18 recites a device for performing the method of claim 23. A device is disclosed by Heile [FIG. 1]. Claim 18 is rendered obvious over Heile in view of Hoskins for rationale similar to that shown above regarding claim 23.

Regarding claim 20, Heile teaches that the management unit defines a plurality of parts of a project for designing or configuration the process control system, each part comprising a given subset of the project design blocks, wherein the project parts are differently assigned to at least two of the programming devices [*“Step 504 determines whether file A is currently locked in the global work space. If file A is currently locked, this indicates that another user is currently modifying file A.”* (column 18, lines 35-65)].

Regarding claim 21, Heile teaches that the software tool protects certified copies of the design blocks in a given programming device from being inadvertently updated by canceling the references to the certified copies while maintaining references to any of the project design blocks required for expansion of the process control system, wherein current project design blocks are

only copied from the management unit to the given programming device that are needed for the expansion [*"Step 504 determines whether file A is currently locked in the global work space. If file A is currently locked, this indicates that another user is currently modifying file A. In this situation, the requesting user may not check out file A and in step 506 an error message is returned and the procedure is done."* (column 18, lines 35-65)].

Regarding claim 22, Heile teaches a software routine on the management unit that transmits a user prompt to all of the programming devices when any of the programming devices requests an updated one of the project design blocks, wherein the software routine requires acceptance of the updated design block by a user of each of the programming devices via user input in response to the user prompt, wherein after receiving said acceptance from all of the programming devices the software routine transmits said updated one of the project design blocks to all of the programming devices, wherein a given revision level for said one of the project design blocks is synchronized on all of the programming devices [*"FIG. 14 is a flowchart 600 illustrating a technique by which a source file may be automatically retrieved from the global work space to a user's local work space. In step 602, a triggering event is received for the automatic retrieval of project source file A from the global work space to a user's local work space. A trigger event may take a wide variety of forms. By way of example, a triggering event ... may occur upon notification from the global work space that file A has changed. Once the trigger has been received, then in step 604 it is determined whether the file A is in fact in a default state for that user and that a new flag has not been set for file A. If file A is not in a default state for a particular user, then that file will not be automatically retrieved. [...] If the*

Art Unit: 2123

*result is YES from step 604, then in step 606 it is determined whether the file A version in the global work space is a newer version than file A version in the user's local work space... If the results is YES, then in step 608 file A from the global work space is copied automatically to the user's local work space. In this fashion, a local project source file in a default state is automatically updated with recent versions of that file from the global work space." (column 18, line 66 – column 19, line 26)].*

***Conclusion***

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jason Proctor whose telephone number is (571) 272-3713. The examiner can normally be reached on 8:30 am-4:30 pm M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Paul Rodriguez can be reached at (571) 272-3753. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Jason Proctor/  
Examiner  
Art Unit 2123

jsp